云原生发展白皮书

（2020 年）

云原生产业联盟

Cloud Native Industry Alliance, CNIA

2020 年 7 月

版权声明

编写说明：

牵头编写单位：中国信息通信研究院
参与编写单位：阿里云计算有限公司、百度云计算技术（北京）有限公司、北京凌云雀科技有限公司、北京金山云网络技术有限公司、华为技术有限公司、杭州谐云科技有限公司、上海道客网络科技有限公司、苏州博纳讯动软件有限公司、腾讯云计算（北京）有限公司、浙江蚂蚁小微金融服务集团。

编写组成员：

中国信息通信研究院：栗蔚、陈屹力、刘如明、闫丹、郑立
阿里云计算有限公司：易立、李小平、朱松、李鹏、石兵、阚俊宝、王炳燊、黄玉奇、张大江
百度云计算技术（北京）有限公司：周岳骞、曹剑
北京凌云雀科技有限公司：刘嘉伟
北京金山云网络技术有限公司：赵琦
华为技术有限公司：刘赫伟、马达、张琦、王泽锋、赵华
杭州谐云科技有限公司：王翱宇、才振功、方佳伟
上海道客网络科技有限公司：郭峰
苏州博纳讯动软件有限公司：伞亚鹏、刘欣雨
腾讯云计算（北京）有限公司：罗茂政、邹辉、韩欣、任秀森、陈一苇、王玉君
浙江蚂蚁小微金融服务集团：宋净超

注：编写单位按首字母顺序排列

前　言

当前全球的数字化浪潮逐步加深，云计算成为当今信息化发展的重要基础设施，云原生在数字化浪潮中的角色逐步提升，成为业务创新发展的重要驱动力。

本白皮书是继《云原生技术实践白皮书（2019）》之后，针对国内云原生产业发展现状进行梳理，从云原生概念到新技术发展特征，以及到云原生支撑行业领域发展，最后再总结云原生未来发展新趋势。

目 录

一、 新机遇下的云原生

(一) 重新认识云原生

云原生成为近几年云计算领域炙手可热的话题，但业界普遍存在对云原生概念理解不清晰、内涵认知不统一的问题，为了更好的推广云原生理念，信通院针对云原生概念进行重新梳理，重点从产业效用、技术特征和应用价值三个方面进行深入剖析，以帮助不同领域的受众群体更好的理解云原生，进一步推进国内的云原生产业发展和落地实践。

从产业效用方面来看，**云原生极大的释放了云的红利**，云原生充分继承云的设计思想，未来应用将更多基于云上进行本土应用开发，即云原生应用更加适合云的架构，而云计算也为云原生应用提供较好的基础支撑，如资源隔离、分布式、高可用等，云原生最大程度发挥了云的优势。**云计算的拐点已至，云原生成为驱动业务增长的重要引擎**。云计算的发展已进入成熟期，云原生作为新型基础设施支撑数字化转型的重要支撑技术，逐渐在人工智能、大数据、边缘计算、5G 等新兴领域崭露头角，成为驱动数字基础设施的强大引擎。伴随全行业上云的逐步深化，企业云原生化转型进程将进一步加速。

从技术特征方面来看，云原生技术架构具备以下典型特征：**极致的弹性能力，**不同于虚拟机分钟级的弹性响应，以容器技术为基础的云原生技术架构可实现秒级甚至毫秒级的弹性响应；**服务自治故障自愈能力，**基于云原生技术栈构建的平台具有高度自动化的分发调度调

谐机制，可实现应用故障的自动摘除与重构，具有极强的自愈能力及随意处置性；**大规模可复制能力**，可实现跨区域、跨平台甚至跨服务商的规模化复制部署能力。

从应用价值方面来看，**异构资源标准化，**容器技术有效解决了异构环境的部署一致性问题，促进了资源的标准化，为服务化、自动化提供了基础；**加速数字基础设施升级解放生产力**，降低用户数字化技术的使用门槛，提高资源的复合利用率，变革研发运营的生产方式，打破组织壁垒，实现研发与运维的跨域协同，提升交付效率，解放生产力；**提升业务应用的迭代速度，赋能业务创新。**云原生技术实现了应用的敏捷开发，大幅提升交付速度，降低业务试错成本，高效响应用户需求，增强用户体验加速业务创新。

云原生是面向云应用设计的一种思想理念，充分发挥云效能的最佳实践路径，帮助企业构建弹性可靠、松耦合、易管理可观测的应用系统，提升交付效率，降低运维复杂度。代表技术包括不可变基础设施、服务网格、声明式 API 及 Serverless 等。

(二) “新基建”开启云原生的新篇章

数字“新基建”带来云计算的空前机遇。2020 年的新冠疫情已经给中国经济发展带来较大冲击，经济下行压力增大，然而数字经济展现出强大的抗压能力，极大程度上对冲了疫情影响。后疫情时代随着企业全面复工复产的稳步推进，“上云用数赋智”已经成为企业共识，企业上云进入攻坚期，数字化转型进程显著提速。同时为提振国内经

济，舒缓疫情压力，中央加紧规划部署“新基建”相关工作，高频发布相关政策，截止 3 月 6 日各省公布的 2020 年重点项目投资计划约 40 万亿元，其中“新基建”成重点方向。空前的市场需求刺激和政策利好引爆数字技术的资本市场，以云计算为核心的新一代数字技术迎来新的发展机遇。

后云计算时代的需求从资源优化转向效能提升。数字化转型大潮下的企业面临着新旧商业形态的剧变，颠覆和重构时刻都在发生。更加快速的感知用户侧的需求变化并做出调整，才能在竞争中持续积累优势。业务的敏捷、弹性、个性化和智能化需求凸显，应用的交付模式也发生深刻变化，轻量化、松耦合、灵活弹性的敏捷技术架构成为主要方向。将支撑业务应用的通用技术模块化、系统化逐渐下沉至云平台，低心智负担且功能丰富的应用支撑能力成为云的突出需求，云计算服务重心逐渐上移。

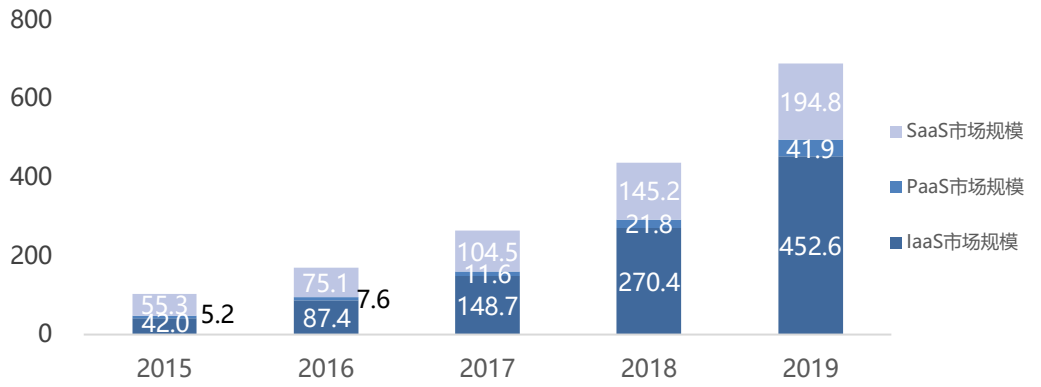
云原生成为下一代云计算的技术“内核”大幅提升用云效能。云原生技术栈统一的标准化交付能力大幅提升云端效能。**服务架构标准统一**，应用微服务化开发，服务之间使用标准的 API 接口进行通信。松耦合架构会减轻因需求变更导致的系统迭代成本，为多团队并行开发提供基础，并加快交付速度；**交付标准统一**，标准容器化的打包方式实现了真正的应用可移植性，不在受限于特定的基础架构环境。并且容器技术进程级的资源切分粒度，也会降低系统的资源开销；**研运过程标准统一**，通过引入 DevOps 理念强化软件研发运营全周期的管理，从软件需求到生产运维的全流程改进和优化，结合统一工具链，

实现文化、流程、工具的一致性，降低应用软件高频发布带来的风险，提升软件产品质量。

二、 云原生产业规模持续看涨，生态版图快速扩张

(一) 云原生产业规模分析

2019 年我国公有云 PaaS 市场规模继续保持高速增长，市场规模为 41.9 亿元，同比增长 92.4%。私有云市场规模为 645.2 亿元，同比增长 22.8%[1]。云原生产业作为现阶段云计算 PaaS 市场的重要支点，也延续了高速增长态势，根据中国信息通信研究院相关调研数据显示，**2019 年我国云原生产业市场规模已达 350.2[2]亿元**。数字经济大潮下传统行业的数字化转型成为云原生产业发展的强劲驱动力，“新基建”带来的万亿级资本投入，也将在未来几年推动云原生产业的发展迈向新阶段。



数据来源：中国信息通信研究院，2020 年 5 月

图 1 中国公有云细分市场规模及增速

[1] 数据来源：中国信息通信研究院《云计算发展白皮书（2020）》
[2] 数据来源：中国信息通信研究院《中国云原生用户调查报告 2020》

我国云原生产业发展已呈现几个明显特征：**大中型互联网企业主导云原生产业发展，技术应用快速向垂直行业扩展**。超 6 成的云原生技术用户为互联网企业，其中千人以上企业规模的企业占比高达 35.11%[3]，互联网头部企业在云原生产业发展中举足轻重。同时金融、制造、服务业、政务、电信等垂直行业的应用占比有所攀升，行业数字化转型的带动效应初步显现；**云原生技术在传统领域尚处发展期 IT 技术投入占比较低，技术研发是资金流出的主要方向**。80%的调研企业在云原生技术领域的投入不足整体 IT 投入的 30%[4]，云原生架构尚未成为企业数字基础设施建设的核心支柱，有近 7 成的用户表示投入的资金主要用于相关技术的研发；**云原生架构采纳用户的生产集群以中小型规模为主，规模化应用仍存在较高技术门槛**。云原生技术栈在规模化应用时的安全性、连续性及性能等因素成为用户侧落地的主要顾虑（调查占比为 61%），同时陡峭的学习曲线以及与现有平台的整合演进也成为用户摇摆的重要因素（调查占比分别为 47%和 46%[5]）。

(二) 云原生产业生态分析

2019 年 Gartner 在容器报告中预测，到 2020 年将有 50%的传统老旧应用被以云原生化的方式改造，到 2022 年将有 75%的全球化企业将在生产中使用云原生的容器化应用。云原生已成为新常态，容器化需求从行业头部企业下沉到中小规模企业，从领先企业尝鲜变为主

[3] 数据来源：中国信息通信研究院《中国云原生用户调查报告 2020》
[4] 数据来源：中国信息通信研究院《中国云原生用户调查报告 2020》
[5] 数据来源：中国信息通信研究院《中国云原生用户调查报告 2020》

流企业必备。随业务的发展，服务器规模相较去年却有了一定的缩减，这一定程度上是容器化带来的效果。K8S 拥有极高的扩展性、自动化和可伸缩性，通过容器化改造，可以最大化地利用服务器资源，按需使用，有效节约服务器成本。

在降本增效的目标下，加快数字化业务发展成为传统企业的必然选择。由于业务场景、用户习惯迅速变化，许多行业数字化业务出现急速增长。数字化业务意味着大刀阔斧的企业敏捷文化，只有借助更加快速、灵活的开发和交付模式，才能满足市场快速变化的需求。

通过使用容器、Kubernetes、DevOps、微服务等这些年轻且先进的技术，能够大大加快软件开发迭代速度，提升应用架构敏捷度，提高 IT 资源的弹性和可用性，帮助企业客户加速实现价值。对于大部分传统企业而言，云原生 PaaS 平台是未来企业业务的核心竞争力的底层支撑，而非核心竞争力本身所在。企业应该将更多的人员、精力和成本投入到与业务相关的研发上，而不是重复造底层基础设施的轮子，造成重复性浪费投资。对于云原生生态中的软件开发商、行业解决方案商等各个领域的合作伙伴也是如此。

最终用户是推动产业发展的原生需求，不同的企业在基础设施和应用架构方面都有自身的个性化差异、任务复杂性等挑战。有的企业只关注云原生中的某一项技术，解决某个切肤痛点。有的企业则希望同时考虑完整体系化的解决方案，追求云计算技术生产力的最大化。每家企业的云原生之路都应该是个性化、量身定制，而且循序渐进的。中国率先控制住疫情为国内企业在数字化转型方面争取到窗口期，数

字化转型在疫情和后疫情时期越发重要。企业上云已经成为一种必然趋势。疫情之下，虽然各行各业都受到了不同程度的影响，但那些数字化能力健全的企业抵御风险的能力更强。

生态伙伴是联接产业供需的重要纽带，最终用户的 IT 应用开发形式（自研或和第三方技术厂商合作）决定云原生生态的格局。除互联网企业外，对于更多中大型企业来说，经过仔细调研会发现云原生技术难度和自研成本很高，大部分企业需要和专业技术厂商合作，共同落地云原生技术，打造技术中台。专业的技术厂商能够提供完善的咨询服务、解决方案和方法论。同时云原生技术的部署也一定程度上伴随着对企业 IT 文化、流程的变革，也需要技术厂商和企业的配合。传统软件服务体系，纷纷与云原生技术对齐，构成云原生生态合作，其中典型代表有:

独立软件开发商，长期聚焦某些行业或者垂直领域，在技术、产品和客户方面都有深厚积累的企业，有能力依托云原生核心技术和平台优势，开发可规模推广的行业产品或解决方案。

软件集成商，具备系统集成资质，有一定的客户基础和方案整合能力，为企业提供优质的云原生解决方案和服务。

IT 服务与交付厂商，有企业信息化实施、服务和行业客户交付能力，通过参与云原生产品解决方案完善、项目咨询、项目实施、项目交付等工作，为企业提供服务。

(三) 云原生技术生态分析

云原生的理念经过几年发展，不断丰富、落地、实践，云原生已经渡过了概念普及阶段，进入了快速发展期。云原生技术以其高效稳定、快速响应的特点驱动引领企业的业务发展，帮助企业构建更加适用于云上的应用服务。过去几年中，云原生关键技术正在被广泛采纳，如 43.9%[6]的用户已在生产环境中采纳容器技术，超过七成的用户已经或计划使用微服务架构进行业务开发部署等，这使得用户对云原生技术的认知和使用进入新的阶段，技术生态也在快速的更迭，特征明显。

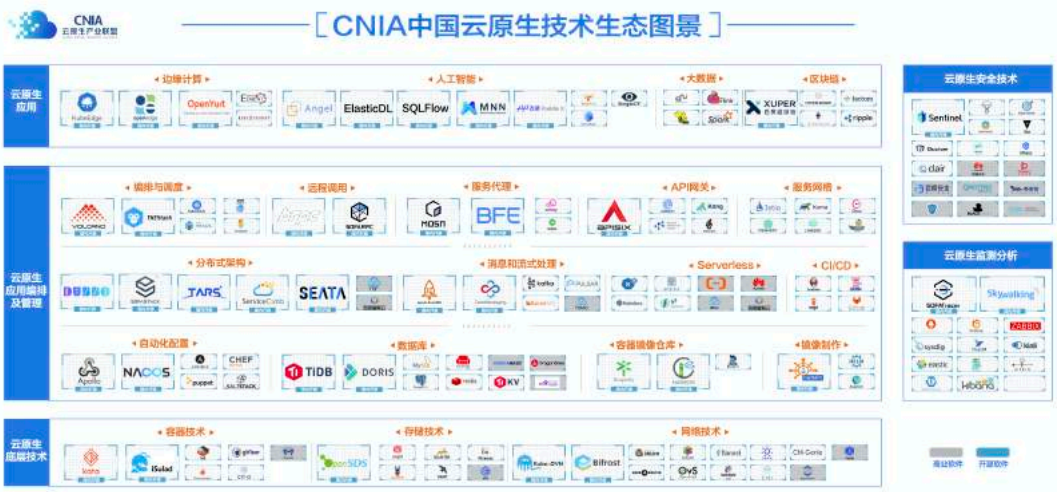


图 2 CNIA 中国云原生技术生态图景

云原生技术生态日趋完善，细分项目不断涌现。相较于早年的云原生技术生态主要集中在容器、微服务、DevOps 等技术领域，现如今的技术生态已扩展至底层技术、编排及管理技术、安全技术、监测分析技术以及场景化应用等众多分支，初步形成了支撑应用云原生化构建的全生命周期技术链。同时细分领域的技术也趋于多元化发展，如

[6] 数据来源：中国信息通信研究院《中国云原生用户调查报告 2020》

在容器技术领域，从 docker 这种通用场景的容器技术逐渐演进出安全容器、边缘容器、serverless 容器、裸金属容器等多种技术形态。

开源技术主导生态，本土力量日益凸显。“软件即服务”的概念逐渐深入人心，在云计算服务化转型的大趋势下，软件开源的经营模式逐渐成为主流，开源的价值也被国内外的公司所深刻认同：开源技术也有助于构建高质量的开发人员社区，实现全球精英人才的协同贡献，加速技术创新的覆盖率指数级增长。鉴于此，全球云计算厂商都在积极布局云原生开源项目，加速企业的技术竞争力。我国云原生技术领域也涌现出大量的国内公司主导的优质开源项目，本土开源力量在支撑国内需求的基础上开始反哺国际社区，如腾讯开源的微服务框架 TARS 贡献给了 Linux 基金会、阿里开源的分布式服务框架 Dubbo 贡献给了 Apache 基金会、容器镜像仓库项目 Harbor 已经从 CNCF（云原生计算基金会）毕业深度应用于企业生产。

三、 云原生热点技术井喷式爆发，细分领域发展趋于多元

(一) 云原生底层技术

1. 云原生服务器：兼顾性能与管理的新算力载体

传统 IaaS 层计算产品形态主要分为裸金属物理机和云服务器两大类。两者在计算性能，管理运维方面各有优势，又都存在不足。云原生服务器则兼顾了物理机的性能优势、完整特性和云服务器的管理便利，进一步解决客户对高性能计算的强需求。

云原生服务器是指基于专用硬件、芯片，利用软硬融合虚拟化等技术将负载或任务转移，提升资源使用效率、用户体验和整体性能的新型服务器。云原生服务器采用软硬一体的硬件卸载和加速技术，通过专用的硬件，将原来在物理机上运行的网络、磁盘、管控等负载，完全下沉到定制的硬件上，物理服务器上的资源可以被最大程度的释放出来，从而提升资源的使用效率，降低成本。同时，通过使用 ASIC 或者 FPGA 等专用芯片来处理存储、网络等任务，可以使用较低的成本将性能提升数倍甚至一两个数量级。此外，软硬融合的虚拟化技术能够支持裸金属形态的计算产品，使其同时具备虚拟机的使用体验和物理机的强大性能。

云原生服务器不仅具有虚拟机的灵活性和弹性，同时具备物理机的一切特性和优势，因此也具备再次虚拟化的能力，而不用担心嵌套虚拟化带来的性能开销。

2. 云原生存储：加速云存储的云原生化改造

云原生存储是云原生应用场景下的存储解决方案，存储形态可为块、对象、文件、键值存储等。云原生存储可以以“声明”的形式被云原生应用申请使用，支持容器编排层对接存储控制面，完成对存储资源的生命周期管理。编排层与存储的交互架构如下图，通过控制面接口直接对接到编排层（下图中 A 所示）为应用负载提供存储资源的动态供应能力，通过 API 框架或者工具间接对接到编排层（下图中 B 所示），提供数据存储的一些高阶能力，如数据保护，数据迁移等。

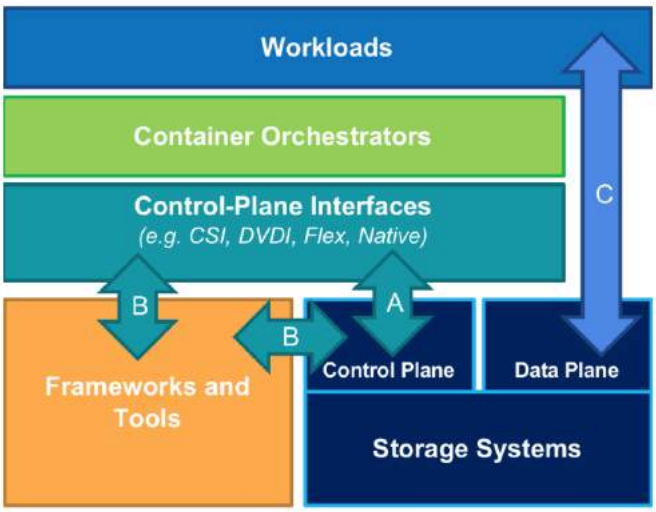


图 3 云原生编排系统与存储交互示意图

Kubernetes 对云存储的原生调度方案难以满足云原生环境下企业核心/智能应用的部署需求。云原生技术已被企业广泛采用，应用容器化运行的比例大幅攀升，使用云原生存储来部署生产可用的有状态应用正呈现加速上升趋势。越来越多的互联网、数据库、消息队列等企业有状态核心应用，逐步迁移到云原生平台，对不同的云上块存储的性能在时延和吞吐，以及稳定性提出新的要求。随着云原生应用对可迁移性，扩展性和动态特性的需求，对云原生环境下的存储也提出了相应的密度、速度、混合度的新要求，对云存储基本能力之上又提出了在效率，弹性，自治，稳定，应用低耦合，GuestOS 优化，安全等方面的诉求。

云原生存储在数据、控制双平面提升存储使用效率与安全性。云原生存储是指通过整合多种云存储形态，完整匹配 Kubernetes 环境下的存储声明机制，提升存储的质量、稳定性及安全等能力，以满足不同应用场景下的存储使用需求的新型存储解决方案。当前主要有两种方式来构建云原生存储能力:

一是对已有存储系统的对接改造。这种构建方法适用于已经具备可用存储系统的公有云或私有云环境，通过标准的 CSI 接口来对接各类存储，通过完善声明式接口和增强原有存储系统能力（扩展性、性能、安全性、稳定性等）来构建云原生存储系统。这种构建方式可以复用云计算基础设施，无需重复发明轮子。**二是面向云原生场景设计全新的存储系统，**云原生存储系统通常基于具体平台存储层（块或者文件）再构建一层分布式存储层，屏蔽了不同平台的存储层差异，提高了应用部署的灵活性，使存储系统可以像云原生应用一样能被部署到任意平台上。

图 4 多场景下的云原生存储选择示意图

3. 云原生网络：面向用户应用场景不断演进

云原生网络的基本目标是满足云原生服务的网络端点和服务间的互通性、安全性和负载均衡要求。Kubernetes 已经成为容器编排的事实标准，容器网络也需与 Kubernetes 的调度机制相匹配。容器网络接口 CNI(Conteinre Network Interface)是现行的网络接口标准，CNI 接口只实现创建、删除容器时的调用方法，其他所有的网络能力

都交由网络厂商实现增值服务，这在一定程度上加速了网络方案的繁荣，但是给用户的方案选型造成了较大困扰。大部分的用户场景都是基于网络的通讯协议进行方案选择，根据网络协议的不同，可将网络方案分为路由模式、Overlay 和 L2 方案三种。

表 1 云原生网络方案对比

	路由模式	Overlay 模式	L2 模式
优点	• 网络性能高 • 支持 Kubernetes 原生负载均衡和网络策略机制 • 符合传统网络的监管要求	• 物理网络无侵入 • 支持 Kubernetes 原生负载均衡和网络策略机制	• 网络性能高 • 可直接与 IaaS 网络层通信，易于迁移 • 符合传统网络的监管要求
缺点	• 大规模应用场景需要交换机与 BGP 打通	• 存在封装影响性能 • 排查问题难，需引入额外排查工具 • 无法与传统的网络监管模式兼容	• 网络管理依赖于物理网络 • 大部分方案无法复用 Kubernetes 的网络优势
实现技术方案	• Calico BGP • Flannel Host Gateway • Kube-router • Contiv BGP	• Calico IPIP,VXLAN • Flannel VXLAN, UDP • WEAVE • Canal • SDN 方案	• Lunix Bridge • Macvlan • SRIOV • OVS Bridge • Contiv Vlan • Ovn-kubernetes

数据来源：中国信息通信研究院

自 CNI 标准发布到 2020 年，云原生网络已经演进近 6 年时间。也积累了大量的用户落地案例和大规模的实践案例。未来对于云原生

网络的演进，依旧会在用户落地场景方向上深度演进。总结起来主要是以下几个趋势:

大规模、复杂的互访场景要求云原生网络扁平化。随着云原生技术的普及，容器集群规模快速增长，跨集群、跨 VPC 互访场景越来越丰富，这要求容器端点具有与宿主节点相同的互通能力，容器和服务具有独立 VPC 的子网地址，甚至具有独立的直通网口,这样在获得更高转发性能、更低损耗的同时，兼顾更好的隔离性。通过在容器挂接的网口配置安全组规则，能够实现容器级别的微分段网络管控策略。但是,容器端点规模和发放速度相对于现有 VPC 网络规格存在数量级的差距，规模扩展问题仍有待解决。

eBPF 等技术将有效改善容器网络复杂链路的高延时问题。容器网络中大量依赖了 Linux 的网络虚拟化的技术，例如 iptables、bridge 等，这些复杂的链路导致网络延时显著增加。而在 Linux 新版本内核中引入的 eBPF[7]技术可以通过可编程的方式去简化内核的网络转发链路，通过把 XDP 程序注入到了网卡驱动程序中，大幅度缩短网络处理链路，降低复杂度，提升了网络的可靠性和性能，在未来会有广泛的应用。

网络安全将成为云原生技术底座的重要组成部分,平台的安全问题在所有的平台演进和建设过程中一直扮演着非常重要,但是不十分紧急的角色，在容器安全建设上，大部分组织都是采取防守和被动姿态。但是本身在近几年陆续爆出大量的基于容器平台的安全隐患以及

[7] eBPF：Extended Berkeley Packet Filter，Linux 内核中的一种报文过滤机制

在国内“护网行动”的大背景之下，容器安全已经成为云原生底座无法绕开的一个问题，容器网络安全在整个底座安全里面扮演了非常重要的角色，也将成为之后的 CNI 网络演进的方向和趋势。

云原生网络的规模扩展问题仍然有待解决。容器网络和 VPC 网络的扁平融合的趋势之下，容器端点规模和发放速度相对与现有 VPC 网络规格存在数量级的差距，单节点的弹性网口密度和弹性扩容速度依然不能满足云原生工作负载的要求，规模扩展问题仍待解决。

4. 容器技术：从通用向多元化发展

（1） 安全容器

容器技术的采纳率连年提升，已经开始进入企业的生产环境。以 Docker 为代表的普通容器通过 Namespaces 和 cGroups 实现的隔离，共享内核的机制使得隔离性具有天然的缺陷无法根除，在多租户场景下安全问题更加凸显：

内核 Bug 引发容器逃逸，操作系统内核漏洞、Docker 组件设计缺陷、不当的配置等都会导致 Docker 容器发生逃逸。由于频发的安全及逃逸漏洞，一般在云环境中的容器应用不得不运行在虚拟机中，以满足多租户安全隔离要求。而分配、管理、运维这些传统虚拟机与容器轻量、灵活、弹性的初衷相悖，同时在资源利用率、运行效率上也存在不足。**内核资源竞争影响业务性能**，同一个宿主机上的不同 Pod，实际上是不同的用户态进程的集合，这些用户态进程虽然在 namespace 上是相互隔离的，但他们还是会共享很多内核资源，比如

调度器、某些内核线程或者对象。这种级别的资源共享会引入很多可以观测到的性能抖动，对在线业务的影响也很明显。

与 Docker 普通容器不同，安全容器通过添加隔离层，给进程分配了一个独立的操作系统内核，从而避免了让容器共享宿主机的内核。因此容器进程能够看到的攻击面，就从整个宿主机内核变成了一个极小的、独立的、以容器为单位的内核，从而有效解决了容器进程发生“逃逸”或者夺取整个宿主机的控制权的问题。

（2） Serverless 容器

FaaS 平台提供的是函数级别的 Serverless 化部署，且应用场景多依赖于其绑定的触发器，对函数的执行有一些配置限制，并且不支持进程常驻。传统的应用大都是单体应用或者微服务应用，在迁移到 FaaS 平台时，需要拆分函数，迁移成本较高。

Serverless 容器，可以很好地弥补 FaaS 的不足，Serverless 容器可以支持进程常驻的服务形态，不限运行时长，并扩大 Serverless 的应用场景。Serverless 容器支持服务的形态，传统的单体应用或者微服务应用，几乎可以无缝迁移到 Serverless 容器平台上。

Serverless 容器和传统的容器相比，为了实现 Serverless 的理念，在如下几个方面做了加强：**免运维的纯托管模式**，传统的容器往往是直接将容器集群托管给业务方，业务方需要分担容器集群的一些运维工作。Serverless 容器则把容器集群完全托管给云厂商，由云厂商进行集群的运维工作，用户不用关注这些运维工作，只需部署自己

的业务逻辑即可；**以实际资源用量计费**，传统的容器是按照容器的实例配置进行计费的，Serverless 容器是按照实际资源使用量进行计费；**秒级弹性伸缩响应**，传统的容器往往借助于容器编排工具来实现弹性伸缩，比如通过 Kubernetes 可以实现 Docker 的容器的弹性伸缩，但是 Kubernetes 伸缩时间是分钟级的，而 Serverless 容器能够提供更加极致的伸缩能力，做到秒级伸缩并且资源实例和伸缩至零。

（3） 裸金属容器

容器服务最早部署形态是基于 IAAS 虚拟机，以虚拟机节点作为容器集群的计算节点，并基于此构建容器的网络、存储和编排能力，这样的堆叠架构虽然可以让整个软件栈分工明确、边界清晰，但是带来了较大的性能损耗和功能冗余。此外如果用户对实例安全隔离性要求较高，就需要借助虚拟化技术，而虚拟化平台不能很好支持该能力。基于以上痛点，在裸金属服务器上搭建容器服务成为一些对性能和实例隔离性较高用户的选择。

随着裸金属容器的发展，为了进一步提高容器负载性能和稳定性，原来部署在裸金属之上的非业务负载组件也逐步的由专门的卸载硬来承载，比如容器存储、容器网络、容器引擎以及服务网格组件。将容器组件下沉到卸载卡后，有几方面好处:

- 裸金属节点就可以被当做纯粹的计算资源，可以“完全”被业务负载使用。同时避免了对业务负载的性能干扰。

- 容器网络、容器存储组件下沉到卸载卡后可以与传统 IAAS 层的网络、存储组件垂直打通，减少冗余功能；直接以硬件设备直通方式将存储、网络资源分配给容器实例，缩短 I/O 路径，提高性能。
- 容器层组件下沉到卸载卡后，裸金属成为纯粹的计算资源，可以被容器实例或者虚机实例共享，为虚拟机和容器实例共节点奠定了基础，提高资源整体利用率。

虽然裸金属容器可以通过卸载技术获得诸多益处，但同时也面临着较大的挑战:

- 资源占用问题。由于卸载卡上的资源非常有限，容器组件需要进行轻量化瘦身后才能较好的适配卸载卡，当前业界也在推动容器引擎层面的轻量化改造，比如 kata-shim-v2 和 isulad。
- 实例密度问题。由于容器存储和网络资源都是走 VF 直通方式，而当前卸载卡上支持的 VF 数量比较有限，在小规格实例场景，VF 会成为实例密度提升的限制。

此外，裸金属容器不仅在资源利用率和性能上有优势，对系统运维管理的自动化和敏捷性上有较高诉求。为了获得较高的自动化运维能力，很多的依赖组件都进行了微服务改造，借助容器编排自身能力来自动化管理所依赖的服务，甚至是节点操作系统本身。比如 AWS 为了提高容器计算节点操作系统更新管理的灵活性，推出了 bottlerocket 产品，放弃原来基于包更新的升级机制，采用镜像粒度

一步更新方法，降低了 OS 更新的失败率，提高运维自动化程度和容器应用的稳定性。

(二) 云原生编排及管理

1. 云原生消息队列

消息队列是指利用高效可靠的消息传递机制进行与平台无关的数据交流，并基于数据通信来进行分布式系统的集成。传统应用架构设计中**系统组件与应用紧耦合**，消费者出现任何问题（升级停服、宕机、不可用等），都会影响生产者的业务；**系统可用性和效率低**，突发的海量消息压力，消费者无法实时高效的处理消息时，容易产生雪崩效应；**缺少持久化机制**，系统发生故障会丢失消息；**消息本地存储难扩展**，单机的处理能力和内存容量都是有限的，不具备可扩展性，同时系统组件高度耦合，扩展难度大。

为解决传统架构中的种种问题，云原生消息队列服务应运而生，它为微服务和事件驱动架构提供核心的解耦、异步和削峰的能力。通过消息队列能够让用户很容易架构出分布式的、高性能的、弹性的、鲁棒的应用程序。

程序组件与应用解耦分离独立运行，同时还可以简化组件间的消息管理。分布式应用程序的任何组件均可将消息存储在队列中， 云云消息服务 确保每条消息至少传送一次，并且支持多次读取和写入。单个队列可由多个分布式应用程序组件同时使 用而无需这些组件之

间的互相协作。所有组件均可使用云消息服务 API 以编程方式检索和操作消息。

可靠的基于消息的异步通信机制，能够将分布式部署的不同应用（或同一应用的不同组件）之间的收发消息，存储在可靠有效的 消息队列中，防止消息丢失。云原生消息队列支持多进程同时读写，收发互不干扰，无需各应用或组件始终处于运行状态。

消息同步多副本落盘保障消息高可靠，通过分布式 Raft 等算法保证消息强一致，提供消息队列、发布订阅、消息回溯、延时消息、顺序消息、消息轨迹等服务。具有高可靠、高可用、高性能、动态伸缩等优势。

云消息服务不仅具备处理系统解耦，异步通信等传统消息队列所必备的能力，而且在数据的可靠传递，性能，快速部署等方面提供有力的支持，满足使用者针对特定场景下的消息的高可靠堆积，动态扩缩容，系统监控，消息轨迹查询等方面的需求。

2. 服务网格

服务网格 （Service Mesh）是一个用于管理、观测、支持工作负载实例之间安全通信的管理层。服务网格通常以轻量级网络代理阵列的形式实现，这些代理与应用程序代码部署在一起，而对应用程序来说无需感知代理的存在。服务网格通常由控制平面和数据平面两部分组成。数据平面运行在 Sidecar 中，Sidecar 作为一个独立的容器和业务系统运行在同一个 Kubernetes 的 Pod 里面，或者作为一个独

立的进程和应用程序进程运行在同一个虚拟机上，其主要充当业务系统的网络流量的代理。传统 RPC 中的服务发现、限流、熔断、链路追踪等能力都会下沉到 Sidecar 中。Sidecar 为应用程序提供了一个透明的网络基础设施，让业务在低侵入或者零侵入的情况获得更健壮的网络通信能力。

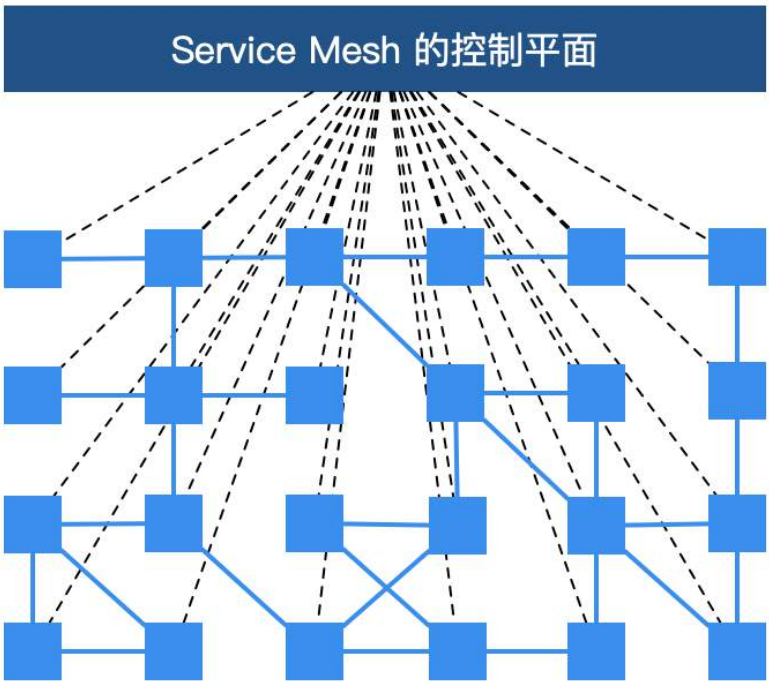


图 5 服务网格控制平面示意图

服务网格为微服务带来新的变革，主要体现在：**服务治理与业务逻辑解耦，**服务网格把 SDK 中的大部分能力从应用中剥离出来，拆解为独立进程，以 Sidecar 的模式部署，将服务通信及相关管控功能从业务程序中分离并下层到基础设施层，使其和业务系统完全解耦，使开发人员更加专注于业务本身；**异构系统的统一治理，**通过服务网格技术将主体的服务治理能力下沉到基础设施，可方便地实现多语言、多协议的统一流量管控、监控等需求。

服务网格相对于传统微服务框架拥有较多优势：**超强的通信线路数据观察性，**服务网格是一个专用的管理层，鉴于它在技术堆栈中处于独特的位置，所有的服务间通信都要经由服务网格，以便在服务调用级别上提供统一的遥测指标；**“面向目的地的”流量控制能力，**由于服务网格的设计目的是有效地将来源请求调用连接到其最优目标服务实例，所以这些流量控制特性是“面向目的地的”，而这也正是服务网格流量控制能力的一大特点。通过服务网格，可以为服务提供智能路由（蓝绿部署、金丝雀发布、A/B test）、超时重试、熔断、故障注入、流量镜像等各种控制能力；**微服务网络的增强安全特性，**在某种程度上，单体架构应用受其单地址空间的保护。一旦单体架构应用被分解为多个微服务，网络就会成为一个重要的攻击面，更多的服务意味着更多的网络流量，这对黑客来说意味着更多的机会来攻击信息流，而服务网格恰恰提供了保护网络调用的能力。服务网格的安全相关的好处主要体现在以下三个核心领域：服务的认证、服务间通讯的加密、安全相关策略的强制执行。

服务网格带来了巨大变革，拥有其强大的技术优势，被称为第二代“微服务架构”。然而软件开发没有银弹，传统微服务架构有许多痛点，而服务网格也有它的局限性：**网络复杂性大幅增加**，服务网格将 Sidecar 代理和其它组件引入到已经很复杂的分布式环境中，会极大地增加整体链路和操作运维的复杂性；**学习曲线较为陡峭**，当前的服务网格几乎都建立在以 Kubernetes 为基础的云原生环境上，服务网格的运维人员需要同时掌握 Kubernetes 和服务网格两种技术，

才能应对使用中的问题，增加用户的学习成本；**系统调用存在额外性能开销，**服务网格在服务链路中引入了 Sidecar proxy，因在系统调用中增加了跳转而带来了延迟。虽然该延迟是毫秒级别的，在大多数场景下是可以接受的，但是在某些需要高性能低延迟的的业务场景下，可能是难以容忍的。

3. 无服务器架构技术

基础设施架构总是伴随软件架构演进。单体架构时代应用比较简单，应用的整体部署、业务的迭代更新，物理服务器的资源利用效率足以支撑业务的部署。随着业务的复杂程度飙升，功能模块复杂且庞大，单体架构严重阻塞了开发部署的效率，业务功能解耦，单独模块可并行开发部署的微服务架构逐渐流行开来，业务的精细化管理不可避免的推动着基础资源利用率的提升。虚拟化技术打通了物理资源的隔阂，减轻了用户管理基础架构的负担。容器/PaaS 平台则进一步抽象，提供了应用的依赖服务、运行环境和底层所需的计算资源。这使得应用的开发、部署和运维的整体效率再度提升。无服务器架构技术则将计算抽象的更加彻底，将应用架构堆栈中的各类资源的管理全部委托给平台，免去基础设施的运维，使用户能够聚焦高价值的业务领域。

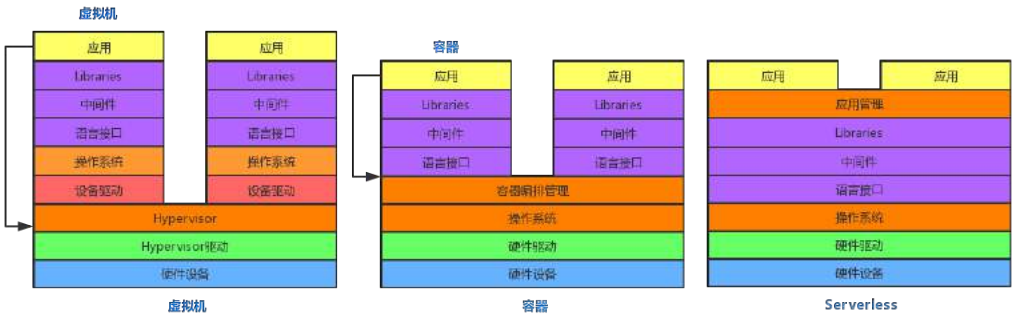


图 6 虚拟机、容器、Serverless 抽象示意图

无服务器是一种架构理念，其核心思想是将提供服务资源的基础设施抽象成各种服务，以 API 接口的方式供给用户按需调用，真正做到按需伸缩、按使用收费。这种架构体系结构消除了对传统的海量持续在线服务器组件的需求，降低了开发和运维的复杂性，降低运营成本并缩短了业务系统的交付周期，使得用户能够专注在价值密度更高的业务逻辑的开发上。

表 2 无服务器架构与传统架构区别

	传统架构	无服务器架构
资源申请	需要申请资源以及后期的资源扩容	无需申请资源，资源按需分配
资源运维	需要运维系统资源，包括资源的容灾，弹性伸缩，安全等	资源的运维交由云厂商代为维护，用户无需运维
计费方式	根据实例配置计费	根据实际资源使用量计费
弹性伸缩	弹性伸缩分钟级，伸缩敏感度不高，不支持缩容到零	更极致的弹性伸缩能力，可以实时/秒级的对资源进行弹性伸缩，并支持缩容到零

在无服务器架构的理念和方法下，有很多种无服务器的技术形态，目前成熟落地的有 3 种形态，函数即服务（FaaS）、后端即服务（BaaS）和 Serverless 容器。

4. 云原生调度系统

随着云原生技术的普及，越来越多的应用开始进行云原生化架构升级和应用迁移，负载类型和集群规模的不断扩大，要求云原生调度系统提升资源的使用效率，并能够对不同类型应用（如 AI、大数据、HPC）、异构硬件资源（如 GPU、TPU）和多个云环境进行统一智能化调度:

离、在线应用的统一调度，越来越多的批量计算任务迁移至云原生环境中执行，从 HPC 到大数据再到人工智能，早期每种场景中的分布式系统多为专有系统。在批量计算任务向云原生环境迁移的过程中，需要云原生调度系统能够支持各种场景的批量计算任务，并能够与微服务应用共享云原生环境和资源，这就需要云原生调度系统要能够同时支持微服务类应用(在线)和批量计算任务(离线)共享资源，以达到多种应用统一调度的要求；**异构硬件的统一调度**，为了应对多种应用对资源适用场景的多元需求，需要云原生调度系统能够对异构硬件资源进行统一的管理与调度，使用各种应用达到最优的资源配比；**多云负载的统一调度**，为降低厂商绑定的风险，同时也为了最大限度的兼顾不同云厂商的优势，多云的环境下的负载高效分发逐渐成为趋势；**智能化的统一调度**，云原生环境的多样化使得资源和作业调度更加困

难，人工智能等新技术的发展为其提供了解决方法。借助作业画像等AI 技术预判资源的使用量，并以此为依据对资源进行动态扩缩容、动态超分，以优化复杂环境下的资源和作业调度方案。

针对上述这些新的要求和挑战，云原生的相关项目提供了不同的解决方案：**一是通过改进现有的 Kubernetes 组件实现**。目前云原生环境中大量的微服务都使用 Kubernetes 进行调度管理，随时着批量计算任务向云原生环境迁移，Kubernetes 社区不同的兴趣组也对各个组件做出相应的改进,调度兴趣组开始通过调度框架，让用户通过自研插件来支持批量计算任务 Kubernetes 社区通过调度框架，希望用户通过自研插件来支持批量计算任务。**二是基于 Kubernetes 的扩展机制实现。**通过 CRD、Operator、multi-scheduler 等扩展机制提供批量计算解决方案，能够有效的支持批量计算任务与微服务应用统一调度，代表性的项目如 Volcano。Volcano 是 CNCF 首个面向批量计算场景下的云原生调度系统项目，作为基于容器组的调度框架，Volcano 支持批量计算的各种场景，并通过引用 Kubernetes 现有的调度策略支持微服务场景并兼容 Kubernetes 的调度策略，同时还支持 GPU 共享的异构硬件功能。

5. 多云容器编排

企业生产环境容器集群规模爆发式增长，越来越多的企业核心业务切换到容器，容器技术栈需要应对的场景越来越复杂，对多云环境的容器编排能力提出了强烈诉求，主要的场景需求如下:

避免厂商锁定，应用可以灵活地部署在不同云供应商或本地 IDC 的集群中，不再依赖某一家云服务厂商；**云上云下分离部署**，部分核心业务部署在私有云环境，满足行业监管和数据安全要求，普通业务部署在公有云上，利用公有云强大的计算能力，同时节约成本；**跨云部署实现容灾**，在云服务商发生故障时可以快速切换到其他的云服务商或者混合云环境中去，实现业务的容灾管理；**跨云弹性伸缩**，利用公有云超大资源池应对短期流量高峰场景，大幅提高业务的承载能力。

当前多云容器编排的实现有两种主要实现方式：**以 GitOps 为媒介联通多云环境的 Kubernetes 集群进行协作**。GitOps 是一种持续交付的理念，其核心思想是将应用系统的声明性基础架构和应用程序存放在 Git 库中进行版本控制。通过将 GitOps 与 Kubernetes 结合，利用自动交付流水线将更改的应用到指定的任意多个集群中。同时，还可通过工具将各集群环境上的实际状态与 Git 库中的配置做比较，快速发现集群配置的非预期变更，并将其还原到预期的状态，以确保多集群环境配置的一致性，从而达到多云容器编排部署的目的。然而 GitOps 方案目前只能解决跨云部署的配置一致性问题，而无法解决跨云应用容灾切换、跨云弹性伸缩等高阶业务场景诉求；**通过集群联邦项目扩展资源类型实现跨集群编排调度**。SIG-Federation 于 2016 年正式推出官方项目 Federation，并在此基础上发展出了 Kubefed 项目。Kubefed 简化了 Federated API 的扩展方式，并在跨集群服务发现和编排等方面做了增强。Kubefed 通过 CRD（Custom Resource Definitions）机制来完成 Federated Resources 的扩充。联邦控制

器则管理这些 CRD，负责把资源分发到不同的集群中，并通过 ExternalDNS 等服务发现机制打通不同集群的应用访问与协同。KubeFed 遵循模块化与可定制的设计理念，并与 K8S 生态保持兼容性与扩展性，由此，基于 KubeFed 的架构理论上可支持扩展解决各类多云场景的问题。

6. 有状态应用管理

以容器为代表的云原生技术，用开放、标准的技术体系，帮助企业和开发者在云上构建和运行可弹性扩展、容错性好、易于管理、便于观察的系统，已经成为释放云价值的最短路径。通过声明式 API 和控制器模式（Controller Pattern），Kubernetes 构建出一套“面向终态”的编排体系，已经成为容器编排的事实标准，为用户提供了敏捷、弹性、可移植性等核心价值，被广泛用于自动部署、扩展和管理容器化应用。无状态应用容器化已是非常普遍地现状，但由于有状态应用基本上都是分布式的，其生命周期管理比无状态应用会复杂很多。

Kubernetes 现有资源类型无法实现有状态应用的合理抽象与描述，有状态应用对外部资源具有一定的绑定性依赖，多个实例之间往往有着拓扑关系，且这些实例本身并不完全等价。Kubernetes 内置的 StatefulSet 资源类型在管理有状态应用方面仅解决了启动顺序、存储状态依赖性，无法实现应用“状态”的合理的抽象与描述，状态和容器进程本身无法解耦；**有状态应用的管理经验无法得到有效沉淀，**在当前环境下，开发者不得不去尝试编写一套复杂的管理脚本，而这

些脚本、知识、和经验，并没有一个很好的办法能够有效的沉淀下来。这一现状制约着云原生应用更广泛的普及。

面向交付及运维的 Operator 技术提供了打包、部署、管理 Kubernetes 应用的全新方式。Operator 一种通过扩展 Kubernetes 的资源模型，引入自定义控制器，实现对资源及其状态的灵活控制的技术。 Operator 把控制器模式的思想贯彻得更加彻底，在 CRD 基础上加上自定义控制器，实现丰富、可扩展的调度及运维功能。Operator 的本质是一段代码，这段代码通过实现 Kubernetes 的控制器模式，来保证这个 CRD 资源始终跟用户的定义完全相同，而在这个过程中，Operator 也有能力利用 Kubernetes 的存储、网络插件等外部资源，协同的为应用状态的保持提供帮助。

7. 云原生数据库

Gartner 在数据库市场分析报告中预测到 2022 年 75%的数据库将被部署或迁移至云平台，只有 5%的数据库会考虑部署在本地。在国外，以 AWS 为主的云厂商已实现云技术与数据库的融合，在逐步挑战与吞噬传统数据库厂商市场份额。在国内，云+数据库的发展也已到了下半场，数据库正在不断深度融合云计算的特性，为用户提供高弹性、分布式、低成本的极致使用体验。数据库的部署形态正在发生改变，传统数据库的发展已遭遇瓶颈:

基于物理服务器部署的传统数据库存在着高成本，低效率，稳定性差等诸多问题。业务部门采购时需为未来业务增量预留资源，存在

着显著的资源浪费。当业务快速增长时，资源的采购周期又无法满足业务快速扩张的需求。此外传统 IT 架构下服务器故障恢复时间较长，对数据库影响明显。**搬迁上云的云数据库，架构层面没有质变无法充分复用云平台的强大能力**，云数据库基于云平台的弹性调度和虚机迁移等能力在一定程度上提高了弹性和稳定性，但受限于传统数据库技术能力限制，这一架构同样存在问题，如较难实现 TB 级数据的存储扩展、指数级增长的并发访问导致数据同步延时严重，可用性降低等。当前解决这些问题的方式主要有业务拆分及采用分库分表的分布式数据库架构两种方式，但以上两种方法同时会带来系统复杂度提高、扩展操作难度大的问题。

云原生数据库的出现能够有效解决云数据库的现有问题，成为重要的发展趋势。云原生数据库采用计算存储分离的架构，遵循“日志即数据”的原则，计算层能够自动实现读写分离，扩缩容过程对上层透明，存储层采用多租户分布式高可用存储系统，单集群具备 TB 级的扩展能力。相对于传统数据库云原生数据库主要有以下几方面优势:

数据库部署高度标准化，云原生数据库本身使用 Kubernetes 等云原生平台进行部署，以 PaaS 的形式对外提供服务，提供近似对象存储的使用体验。这使得数据库无论是在资源使用，还是应用部署，亦或是业务调用上都实现了标准化；**大幅降低使用成本**，与现有数据库架构不同，用户无需提前预留资源，仅需为使用的资源进行付费。受益于计算存储分离架构，增加从库时不会增加存储成本，仅需为计算资源付费；**多数据库版本兼容**，云原生数据库一般兼容现有数据协

议，如 MySQL、PostgreSQL 的多种主流版本及部分商业数据库版本，业务无需修改即可进行迁移。同时数据库提供的 TB 级的空间可满足绝大部分应用需求，免去了分库分表所带来的数据一致性问题，研发仅需要关注业务逻辑即可；**秒级的故障恢复提升稳定性**，传统数据库稳定性严重依赖底层资源，当资源故障时需进行数据迁移，导致故障恢复时间过长。在云原生架构下，计算节点故障时无需拷贝数据，仅需创建新的资源承载流量即可，可做到秒级别的故障恢复。而存储节点一搬采用分布式多副本的架构，其故障更是对业务无感。

(三) 云原生应用

1. 大数据

随着数字化进程的加速，数据逐渐成为各个企业的核心资产，为解决这些数据的存储/分析等问题的大数据也成为企业的基础平台。数据存储及分析需要很强的数据管理能力及资源调度能力。因此，大数据也由原来的存算一体架构向存算分离的架构演进，并进一步向云原生环境演进，借助云原生环境提供的存储及计算能力。但在向云原生环境迁移过程中需要解决以下几个问题:

- **调度:** 在大数据应用向云原生迁移的过程中，一般会由原来的 Hadoop 平台向云原生容器平台 Kubernetes 迁移；但由于 Kubernetes 初期一直以微服务为主，在调度方面还需要有很多加强；比如说常用的公平调度，多级队列以及作业抢占等。

- **数据管理:** 数据是大数据场景下非常重要的一个环节,按照数据的用途，一般可以分为两种: 输入/输出 和 临时数据; 并且大数据多采用存算一体的架构。在云原生的环境中，计算与存储资源将会分别进行管理，以达到降本增效的目标。因此，在大数据应用向云原生迁移中,要将原有的存算一体的架构向存算分离的架构演进; 在演进的过程中，输入/输出数据一般放在对象存储服务中,而计算过程中产生的临时数据则需要大数据架构的支持，例如 Spark 3.0 中引入的 Remote Shuffle Service。
- **网络、本地存储:** 在大数据场景中经常有大量的数据传输，因此需要较高的网络带宽,特别是向云原生迁移而采用存算分离时,带宽显得尤为重要。而支持临时数据的存算分离架构之前，本地存储仍是大数据应用必须的存储资源 (本地存储可以为虚拟磁盘)。
- **加强应用之间的隔离性。** 云原生提倡容器平台托管应用，借助容器，不仅完美的满足了 CPU，内存，网络 IO 等资源的隔离性要求，还支持一次构建随处运行、易于自动化部署运维等特性，同时解耦了大数据应用和底层平台软件依赖，从而提高了大数据应用的部署效率、降低了运维成本。
- **完善的弹性扩缩容能力。** 云原生基础架构提供完善的自动横向扩容和纵向扩容，面对大数据应用的突发流量、随机任务和

离线任务，可以实时扩容满足生产需要，借助云原生完善的监控体系，实现弹性缩容以降低成本和提高资源利用率。

- **统一技术栈。** Hadoop 的 yarn 调度和资源管理方案，专为 Hadoop 生态应用而设计，其功能与当今容器编排领域的事实标准 Kubernetes 存在重叠，然而 Kubernetes 不仅可以支持在线业务，在大数据、AI、边缘计算场景都能够支持，对于企业来说，同时维护两套系统和技术栈，无形增加了研发、运维等运维成本。

2. 边缘计算

万物互联时代加速了云-边协同的需求演进，传统云计算中心集中存储、计算的模式已经无法满足终端设备对于时效、容量、算力的需求，向边缘下沉并通过中心进行统一交付、运维、管控，已经成为云计算的重要发展趋势。针对边缘设备及业务场景的特殊性，边缘应用对容器技术提出了新需求：**资源协同**，边缘计算提供云-边-端的资源协同管理，在云端统一管理边-端的节点和设备，对节点、设备进行功能抽象，在云、边、端之间通过各种协议完成数据接入，在云端统一管理和运维；**应用管理协同，**通过边云协同的方式，将这些编排部署能力延伸到边侧，支持边缘侧日益复杂的业务和高可用性的要求；**智能协同**，AI 能力的发展可谓是近年来边缘计算持续火爆的一大主要推手。边云的智能协同也是目前边缘计算项目中一个非常重要的协同场景；**数据协同，**服务之间的协同更像是要求更高的数据协同。因

为它在数据的传输之外还增加了服务发现、灰度路由、熔断容错等更偏向业务层的能力；**轻量化，**受限于边缘设备的资源，部署在边缘侧的容器平台不可能是完整的 Kubernetes 平台，必须对其进行精简。

以 Kubernetes 为基础，核心价值之一是通过统一的标准实现在任何基础设施上提供和云上一致的功能和体验。在资源协同方面，借助云原生技术，可以实现云-边-端一体化的应用分发，解决在海量边、端设备上统一完成大规模应用交付、运维、管控的诉求。**在安全方面**，云原生技术可以提供容器等更加安全的工作负载运行环境，以及流量控制、网络策略等能力，能够有效提升边缘服务和边缘数据的安全性；**在边缘网络方面**，基于云原生技术的边缘容器能力，能保证弱网、断网的自治性，提供有效的自恢复能力，同时对复杂的网络接入环境有良好的兼容性；**在异构兼容方面**，依托云原生领域强大的社区和厂商支持，云原生技术对异构资源的适用性逐步提升，在物联网领域，云原生技术已经能够很好的支持多种 CPU 架构(x86-64/arm/arm64)和通信协议，并实现较低的资源占用。

以 Kubernetes 为基础的云原生技术和边缘计算相结合，可以很好解决下沉过程中云边一体化协同、安全、边缘网络适配、异构资源适配等难题，极大加速云计算边缘化进程。

(四) 云原生安全

根据 Gartner 研究预测，到 2022 年超过 75％的全球企业将在生产环境中运行容器化应用。虽然以容器为核心的云原生技术发展速度

空前增长，企业采用新兴技术的同时，也需要确保应用在全生命周期的各个关键环节尤其在生产环境运行时的安全问题。云原生技术作为企业数字业务应用创新的原动力，不仅被引入到云原生应用全生命周期管理中，而且被推到了生产环境。云原生技术为企业带来快速交付与迭代数字业务应用的优势之外，同时也带来了新的安全要求与挑战。

传统的边界安全模型在动态变化的云原生环境难适用。云原生环境中应用微服务化大幅增加了内部网络流量和服务通信端口总量，同时承载负载的容器秒级启动或消失的动态变化，增加了安全监控和保护的难度，传统防火墙基于固定 IP 的安全策略很难适应这种持续的动态变化，无法准确捕捉容器间的网络流量和异常行为。

容器共享操作系统的进程级隔离环境增加逃逸风险。传统软件架构下，应用之间通过物理机或虚拟机进行隔离，从安全角度来看可以将安全事件的影响限制在有限可控的范围内。在云原生环境下，多个服务实例共享操作系统，一个存在漏洞服务被攻陷可能会导致运行在同主机上其他服务受到影响，逃逸风险大大提高。

频繁变更对软件流转的全链条安全提出新要求。为提高数字业务应用交付与运维效率，企业应用开发与运维部门引入开发运营一体化流程。每个微服务应用会涉及相对独立的开发、测试和部署的全生命周期，并通过持续集成/持续交付的流水线，将应用部署运行在开发测试和生产环境中。在整个业务应用全生命周期中，需要为各个环节引入自动化安全保护，不仅避免各个环节的潜在风险，而且提高应用安全交付效率。

应用微服务化大幅增加攻击面。容器技术保证了运行环境的强一致性，为应用服务的拆分解耦提供了前提，应用微服务化进程加速，同时也带来新的安全隐患。单体应用拆分导致端口数量暴增，攻击面大幅增加。微服务将单体架构的传统应用拆分成众多个服务，应用间交互的端口成指数级增长，相较于单体应用架构集中在一道口防护的简单易行，微服务化应用在端口防护、访问权限、授权机制等方面的难度陡增。

针对上述云原生技术架构下的安全风险 ，云原生产业联盟 CNIA 联合业内技术专家共同提出云原生技术安全架构模型，用于建设服务之间零信任、具备统一的安全管理策略、软件流转可溯源的云原生安全体系，保证企业的研发运营环境安全。

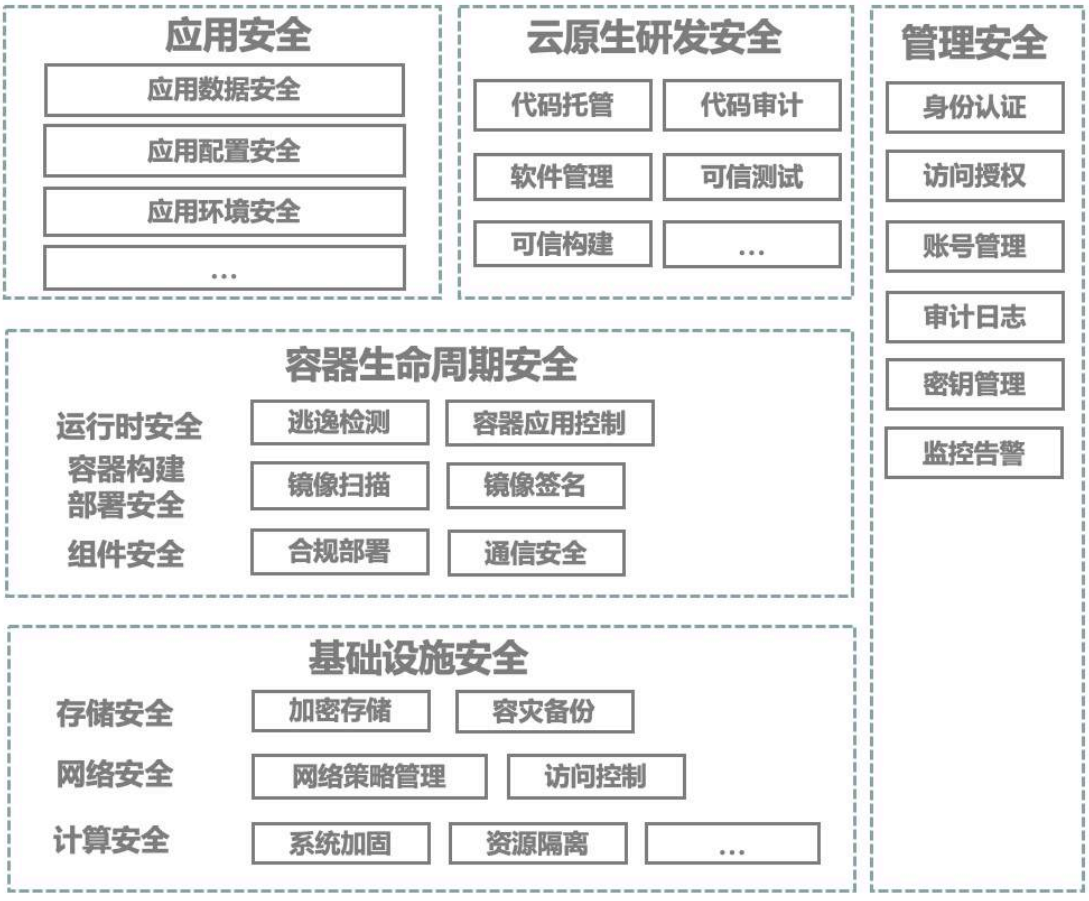


图 7 云原生技术安全架构图

1. 可信计算

可信计算（Confidential computing）的目标是保护数据使用中的安全性和机密性。目前数据的加密和保护技术主要是针对静态存储数据和网络传输数据，这些技术比较成熟。然而如何加密使用中的数据是整个机密数据保护周期中最具挑战的一步。可信计算把加密的数据和代码放在一个特殊的执行上下文环境中，而不会暴露给系统其他部分，任何其他应用、BIOS、OS Kernel、管理员、运维人员、云厂商、甚至除了 CPU 以外的其他硬件均无法访问，极大减少敏感数据的泄露，为用户提供了更好的控制、透明度和隐秘性。

可信计算核心功能包括:

- 保护并验证云端代码和数据的完整性;
- 避免敏代码和数据在使用过程中不被恶意窥探和窃取;
- 保证用户对数据全生命周期的控制。

2. 容器平台安全

容器作为云原生化架构下的重要的资源载体已被被广泛接受和使用，云原生产业联盟的调查数据显示，62%的用户已在生产环境使用容器技术。而在 Sysdig 发布的《2019 年度容器使用报告》数据显示，用户生产环境中 40%的镜像来自公开的镜像仓库，镜像漏洞问题严重，CIS 等安全配置规范在生产环境中落实情况并不理想，容器镜

像、组件安全配置及运行时安全的安全风险极为突出。容器平台的安全防护需要从底层基础设施贯穿至应用。

夯实基础架构安全，基础架构安全作为整个云原生安全框架的基石，需要充分结合云平台底座的平台安全能力，保障平台组件的安全合规，基于云平台 VPC 和安全组功能提供容器网络隔离能力，整合云平台的云安全服务能力，提供云用户和云厂商在 IaaS 层的软、硬件安全防护能力（包括主机安全防护、云数据全链路加密能力、资源访问控制能力和审计能力等）。

强化容器化软件供应链生命周期安全，容器化软件交付是以容器为核心的云原生 PaaS 平台软件供应链的核心，利用持续集成/持续交付流水线，将代码构建为容器镜像，方便的进行进行端到端测试、预发验证和生产部署。保障容器化软件供应链的安全，才能规避潜在安全风险部署上线，并且一旦发现风险，能够可靠评估其影响面，并及时可靠地对其进行修复，软件供应链层的下述安全能力需要重点关注:

- 基于 CVE 的容器镜像扫描和高风险阻断部署能力;
- 基于自定义 ACL 规则的镜像仓库网络访问控制;
- 镜像签名和验签能力;
- 应用交付全链路追踪的可观测性。

构建完善的容器运行时安全检测体系，监控、异常活动检测、隔离是容器应用运行时刻安全防护的关键要素。在容器运行时刻需要保障线上系统可以动态感知系统安全风险，并及时做出响应。这里涉及到容器架构自身的安全性，如容器安全与资源隔离，

Kubernetes 集群安全性，网络策略支持等底层能力。同时需要对容器运行时行为监测，对异常操作、攻击等问题及时告警、阻断。如下特性是容器运行时需要关注的几个关键能力:

- 容器运行时安全防护能力，包括进程异常行为检测、逃逸攻击检测、恶意程序检测等能力;
- 安全沙箱容器，基于轻量化虚拟技术实现容器内核粒度的强隔离;
- 基于网络策略的应用间东西向流量的细粒度访问控制;
- 在多租场景下，通过 PSP 等原生安全特性限制容器运行时的安全风险。对于隔离程度要求较高的场景，支持使用安全沙箱容器和动态策略引擎等强隔离手段保证多租安全。

3. 基于服务网格的安全

在微服务的体系架构下，开发人员基于不同的服务框架和开发语言实现自身服务。分散解耦的微服务环境使得传统安全中的边界防护不在适用，在不改动基础结构的情况下，实现应用安全与现有安全系统的集成形成多层次的深度防御体系，成为云原生安全的探索方向。基于服务网格技术来构建零信任网络的安全解决方案为解决上述问题提供了基础。服务网格技术在微服务安全性方面提供了下列方向上的安全措施:

基于统一规范的认证身份能力，确保服务之间的所有网络流量都是经过安全认证的，在服务应用程序和网关侧提供如 JWT、TLS 双向

认证等多种认证形式，并且支持可配置的认证策略；**全链路的鉴权管理能力**，支持服务与服务间和最终用户到服务间的强制授权检验，确保网格中服务的使用者遵循了权限最小化原则。提供网格维度、命名空间维度和应用负载维度上的基于自定义策略的细粒度授权配置；**服务通信过程的强标识能力**，借由这些标识可实现复杂的访问控制规则定义，对系统行为加以约束。同时结合 mTLS、RBAC 和自动证书轮转等必要安全措施为“零信任”网络的搭建奠定了基础。

四、 云原生价值凸显，加速行业创新应用

(一) 生物医疗

2017 年国家发改委正式印发了《“十三五”生物产业发展规划》，明确了基因检测能力覆盖 50%以上出生人口的目标。利用基因测序、影像、大数据分析等技术手段，在产前筛查方面实现精准的预防、诊断和治疗，成为生物医疗领域的一个重要发展方向，其中以基因检测技术的发展最为迅猛。以云计算为 IT 基础设施已经成为生物医疗行业的重要趋势，但是基因检测行业的特殊性也给计算资源平台带来全新挑战：

业务处理的波峰波谷效应明显，基因检测行业通常在样品达到一定数量后才会进行批量处理，原始数据来源具有较强周期性，业务波动规律明显，资源闲置率高，同时也为用户的资源容量规划增加较多不确定因素；**异构加速硬件环境难统一**，基因检测行业的计算规模庞

大，引入 GPU、FPGA 等异构加速计算硬件提升计算效率是行业刚需，异构加速硬件的环境一致性难以统一，资源的纳管与调度成为瓶颈；**上层应用配置繁琐重复构建效率低**，不同检测类别的应用配置各不相同，频繁切换项目下需反复构建应用场景，手动配置大大降低了生产效率，同时也大幅增加了人为误差的风险。

基于云原生技术栈构建的新一代云平台能有效解决上述行业痛点。**极致的弹性扩缩容能力，有效提升资源利用率**。相比于分钟级伸缩的虚拟机，更加轻量级的容器载体能够在负载的波峰到来时，实现秒级的大规模资源扩展，保障业务资源的顺滑扩容；**服务目录方式提供异构加速硬件驱动增强平台兼容性**。不同加速卡的驱动程序各不相同，新分配资源的驱动安装复杂易错，非常影响用户使用的体验，将驱动程序以服务方式提供，实现一键配置，降低使用门槛，提升平台的兼容性；**标准化封装应用配置简化构建。**借助容器技术将繁琐的配置文件、依赖关系统一打包，实现框架部署的环境一致，极大提升了应用的可移植性，方便用户快速构建业务环境。

(二) 智慧交通

2020 年随着“新基建”对社会治理智能化的推进，智慧社会和民生应用智能化的建设脚步正在大幅提速。智慧交通是民生应用智能化的核心场景之一，随着交通进入信息时代，可持续交通就是既要能够满足不损害当前和未来基本的人类和生态价值的基本要求，又要满足自由交通、获取机会、沟通交往、贸易和建立联系的社会需求。

智能交通运输系统(Intelligent Transportation System，简写ITS)，有别于传统的交通治理、改善技术，它是国际上对运用当代高新科技(计算机、信息、通信、自动控制、电子、系统工程等)提高交通运输效率、增强交通安全性的一系列先进技术或技术集成系统(如交通控制与路线导行系统、车辆行驶安全控制系统、交通运输信息服务系统等)的一个统称。

ITS 在落地的过程中面临着较为复杂的外部条件和多样的需求，使得承载业务的基础设施面临全新挑战：**异构边缘资源的调度利用效率亟待加强**，多种边缘资源的统一接入，包括车辆端、交通基础设施、eNB 基站、MEC 节点等等，需要通过边缘资源的虚拟化，集群管理和智能调度，屏蔽底层异构的特征，提高资源利用率，降低接入门槛；**错综复杂的网络拓扑环境影响了大规模多态实时运算分析的效率**，多种形态的大规模实时数据运算，包括路测设备数据、车辆感知数据、摄像头视频数据等等，需要对海量的数据进行流式处理，并汇聚到云端进行统一存储和分析处理。复杂的网络拓扑环境，不同的边缘设备往往通过不同的网络接入方式连接到云平台，如中心子系统间与路边子系统之间通过电话线、电缆、光纤通信，车-路和车-车通信只能通过红外线、微波、毫米波等无线通信，很大程度上影响了数据处理的效率。

针对 ITS 落地的需求，云原生技术可以帮助服务提供商建设更加适配的分布式基础设施，其中的核心能力包括：

- 大规模集群的分层管理：满足云中心、云边缘、设备边缘多个

环境下的集群资源分层管理的需求;

- 支持第三方集群扩展：满足运营商、解决方案提供商等大量第三方集群的统一接入。
- 轻量级集群控制：集群控制面与数据面解耦，在网络故障或阻塞时不影响边缘业务运行。
- 边缘集群轻量自治：边缘集群可自治管理，满足高延迟敏感业务的就近运算以及边缘集群的资源高效利用。
- 自动容灾：支持跨机房、跨地域的容灾切换，在单点故障时快速专业业务和数据，降低风险。
- 全局统一调度：多集群统一调配资源，分布式编排非延迟敏感型和离线运算业务，最大化优化资源利用率。

通过结合容器化基础设施和微服务化的应用开发方式，云原生技术将会全面赋能智慧交通业务的全面普及，助力智慧交通服务商实现更加全面的感知和控制、更宏大的交通控制平台、更加低廉的实现和普及成本，协助推进“新基建”在民生、交通领域的快速发展。

(三) 工业互联网

2020 年初工业互联网等新基建再次写入政府工作报告，内容从“打造工业互联网平台”，提升为更加“全面的发展工业互联网，推进智能制造”，这也意味着国家对于工业互联网建设和发展赋予了更高使命，从为单个企业赋能，需逐步扩大到为智能制造产业链、为区域经济发展助力，凸显数字经济新优势。

工业互联网面临着各式各样的问题:**种类繁杂的工业资源统一对接纳管难实现**，工业互联网场景中工业 APP 统一调度引擎、工业 APP 底层镜像服务、工业应用 RDB/NDB、第三方能力服务等方面的功能，且需要在平台层面支持多租户和计量计费;**端到端工业应用开发交付闭环难打通**，面向开发者平台需提供开发管理基础能力支撑（如开发工具、开发流水线、制品仓库、代码管理等），知识库组件服务能力（硬件模型库、生产能力库、故障代码库、工业协议库、故障方案库、硬件接口库等服务）、以及应用商店（应用上下架、应用发布等）的能力，这些能力在传统云平台上分化隔离无法形成有机串联的整体。

基于容器和 Kubernetes 技术的云原生平台，为工业互联网应用运行提供灵活敏捷、易于扩展、高可用的运行时环境,降低应用运维的复杂度，提高运维效率。云原生技术栈提供镜像管理、容器管理、集群管理、容器网络、服务编排和负载均衡管理，同时提供多租户隔离、性能监控和一体化管控门户等功能，为工业 APP 统一调度引擎服务、工业应用 RDB/NDB、工业 APP 底层镜像服务、工业知识库、物联网服务、工业应用中间件服务、研发管控服务和第三方能力服务等服务提供运行支撑和服务发布管理，并且提供服务计量计费管理。通过统一集中的资源和配置管理实现对资源的智能配置和应用生命周期管理。**通过 CICD 编排引擎,建立可视化 Pipeline 流水线,打通开发、测试、部署、运维全环节**，实现应用的需求管理、代码审查与代码仓库的管理、自动化测试管理、自动部署、应用升级回滚与上线运行和服务治理等功能。借助微服务思想和 DevOps 理念，把技术、知识、

经验等资源固化为可移植复用的工业微服务组件库,供开发者调用,构建工业应用开发者中心,借助微服务组件和工业应用开发工具,帮助用户快速构建定制化的工业 APP。

(四) 物流

在经济全球化和电子商务的双重推动下,过去十年物流行业发展迅猛,行业规模与业务体量呈现出指数级的增长。千万级的日订单处理量,TB 级的日数据处理量深刻冲击着物流行业头部企业 IT 架构。然而作为传统行业的典型代表,物流行业的 IT 基础架构相对陈旧,应对业务暴增带来种种挑战较为乏力。这些挑战主要表现在:

电商快递促销冲击难扩展,快递业务高峰和电商大促活动同步,原系统所有程序代码都运行在服务器上的同一个进程中,应用程序扩展成本高、难度大,甚至有些应用即便投入昂贵的成本也无法扩展,不能满足业务量快速变化的需求;**业务响应周期长**,原系统编译、部署、上线耗时太长,一个应用从资源申请、环境准备到上线的周期过长,需要大量人力介入,既浪费人力成本,也影响业务整体响应效率;**资源利用率低**,电商大促带来的业务高峰具有明显的错峰特征,物流全环节的多系统流量存在显著的波峰波谷特征;**系统运维难度高**,系统功能代码耦合度高,微小改动可能影响整个应用,应用中心化程度较高,资源无法隔离,系统错误隔离性差,可用性低,任何一个模块的错误都可能造成整个系统宕机。

以数字技术为核心，实现 IT 基础架构的升级以支撑数据驱动的资源管理和物流全过程优化是传统物流行业转型发展的趋势，云原生技术成为 IT 架构升级的重要动力。

通过微服务治理框架、API 网关、全链路业务监控、容器云、DevOps 等云原生核心能力完成平台的云原生化改造能够显著提升基础设施效能。**持续集成持续交付等工具链提升业务自动化程度**，从资源申请、环境准备、测试预发上线到故障修复，一站式解决，且整个过程不再需要大量人力介入，既提升业务整体响应效率，也降低了人力成本；**云原生较强的可观察性提升业务稳定性**，通过服务状态、系统健康度、接口调用情况、异常的实时告警等实现可视化及预警化，自动化的量化和监控功能，结合业务健康检测启用容器级别的异常自动恢复，及时规避业务风险；**应用微服务化改造**，规范服务标准、服务治理以及服务问题追踪，将应用程序代码拆分运行在多个进程之中，简化研发运维，并灵活实现按需水平扩展。

五、 云原生发展趋势

(一) Kubernetes 编排统一化，编排对象不断扩展延伸

在 PaaS 资源编排层面，Kubernetes 已经成为业界公认的事实标准，领先优势非常明显，正呈现出跨领域融合发展趋势。以 Kubernetes 中心的技术、生态日臻成熟和完善。根据云原生产业联盟的 2020 年调查数据显示，Kubernetes 在受访人群的采纳率高达 63%，在容器编

排领域扮演非常重要的角色。**Kubernetes 的编排对象持续丰富不断扩展**，以容器为基础编排对象逐渐延展至虚拟机、函数等，理论上所有可编程、有 API、可抽象成资源的对象，都在成为 Kubernetes 的编排对象。**应用侧围绕 Kubernetes 生态加速演进**，以 Kubernetes 为核心的云原生技术栈将推广到更多的应用场景。在大数据领域，Spark 和 Kubernetes 的集成已经非常普遍；机器学习方面，Kubernetes 和 Tensorflow 等深度学习的框架深度集成，用 Kubernetes 去编排机器学习的工作流以取得业界的广泛共识。

(二) 服务治理 Mesh 化，加速传统应用转型

传统应用架构中业务和功能耦合度较高，无法充分发挥云的效能。传统应用中用于治理服务的中间件服务通常与应用强绑定部署，治理能力被植入每个应用中难以复用，重复造轮子的现象严重。**Mesh 化加速业务逻辑与非业务逻辑的解耦**，将非业务功能从客户端 SDK 中分离出来放入独立进程，利用 Pod 中容器共享资源的特性，实现用户无感知的治理接管。服务治理的 Mesh 化为传统应用轻量化改造提供了前提，也为云平台沉淀通用服务治理能力，加速中间件下沉为基础设施提供了可能。**Mesh 化是传统应用转型云原生的关键路径**，非业务功能的解耦分离使得应用负载大幅减负，更加纯粹的关注业务逻辑。

(三) 应用服务 Serverless 化，更加聚焦业务的核心价值

作为云原生技术未来的演进方向，无服务器架构技术（Serverless）也从观望逐渐落地，据 Gartner 的过往预测数据显示，到 2020 年全球将有 20%[8]的企业部署无服务器架构。Serverless 将进一步释放云计算的能力，**将安全、可靠、可伸缩等需求交由基础设施实现**，使用户仅需关注业务逻辑而无需关注具体部署和运行，极大地提高应用开发效率。同时这个方式促进了社会分工协作，云厂商可以进一步通过规模化、集约化实现计算成本大幅优化。

(四) 从资源云化到业务云化，最终趋于全面云原生化

企业上云的初期阶段是把现有 IT 系统搬迁到云上，更多在虚拟化层面的改造工作，随着云计算生态的蓬勃发展，原有的应用架构陈旧，在扩展性、适配性、弹性伸缩、资源调度、开发运维等方面与云计算架构的优势不匹配，无法真正发挥云的价值。云原生技术通过标准化资源，轻量化弹性调度等特征，应用场景较为广泛，随着技术和生态不断成熟和完善，有效缓解企业上云顾虑，拉动全行业的上云程度。

[8] 数据来源：Gartner 《Top 10 Trends Impacting Infrastructure and Operations for 2019》

中国信息通信研究院

地址：北京市海淀区花园北路 52 号

邮政编码：100191

联系电话：010-62300072

传真：010-62304980

网址：www.caict.ac.cn